March 2003

Decentralized motion planning for multiple robots subject to sensing and communication constraints

Guilherme A. S. Pereira
University of Pennsylvania

Aveek K. Das
University of Pennsylvania

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

Mario F. M. Campos
Universidade Federal de Minas Gerais

Postprint version. Published in *Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II*, Proceedings of the 2003 International Workshop on Multi-Robot Systems, pages 267-278.

Decentralized motion planning for multiple robots subject to sensing and communication constraints

Abstract
We address the problem of planning the motion of a team of mobile robots subject to constraints imposed by sensors and the communication network. Our goal is to develop a decentralized motion control system that leads each robot to their individual goals while keeping connectivity with the neighbors. We present experimental results with a group of car-like robots equipped with omnidirectional vision systems.

Keywords
motion control, dynamic constraints, runtime replanning

Comments
Postprint version. Published in *Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II*, Proceedings of the 2003 International Workshop on Multi-Robot Systems, pages 267-278.

DECENTRALIZED MOTION PLANNING FOR MULTIPLE ROBOTS SUBJECT TO SENSING AND COMMUNICATION CONSTRAINTS

Guilherme A. S. Pereira*, Aveek K. Das, Vijay Kumar†
GRASP Laboratory – University of Pennsylvania, Philadelphia, PA, USA
gpereira, kumar, aveek@grasp.cis.upenn.edu

Mario F. M. Campos
VERLab – Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil
mario@dcc.umfg.br

Abstract We address the problem of planning the motion of a team of mobile robots subject to constraints imposed by sensors and the communication network. Our goal is to develop a decentralized motion control system that leads each robot to their individual goals while keeping connectivity with the neighbors. We present experimental results with a group of car-like robots equipped with omnidirectional vision systems.

Keywords: motion control, dynamic constraints, runtime replanning

1. Introduction

Cooperating mobile robots must be able to interact with each other using either explicit or implicit communication and frequently both. Explicit communication corresponds to a deliberate exchange of messages that is in general made through a wireless network. On the other hand implicit communication is derived through sensor observations that enable each robot to estimate the state and trajectories of its teammates. For example, each robot can observe relative state (position and orientation) of its neighbors (implicit communication), and through explicit communication exchange this information with the whole team in order to construct a complete configuration of the team.

*Guilherme Pereira is grateful for the support of CNPq – Brazil under grant 200765/01-9.
†This work was support by the DARPA ITO MARS Program, grant 130-1303-4-534328-xxxx-2000-0000.

A fundamental limitation related to these forms of interaction among the robots is the limited field of view of the physical sensors and the limited range of transmitters and receivers. When communication is essential to the completion of the specified task, the robots must move in order to maintain such constraints. In this paper, we address the problem of controlling the motion of a team of mobile robots subject to such communication and sensing constraints that we will call *formation constraints*. Formation constraints also arise in other situations. For example, in a object manipulation task, the robots must cooperate in order to keep the object contained in a subset of the configuration space. This requirement on cooperation can be translated into formation constraints that are functions of the object's position and orientation (Pereira et al., 2002). Robots must also avoid collisions with each other. Finally, mapping and tracking tasks may require constraints on relative positions and orientations to guarantee observability (Spletzer and Taylor, 2002).

We will model the team of robots as a set of independent agents that are decoupled except for the formation constraints. Each robot is assigned its own motion plan toward its goal position. The main goal in this paper is to develop a simple strategy for modifying the individual robot motion plans in runtime to maintain the formation constraints. Thus each robot is able to reach its destination while satisfying the formation constraints. While our main focus in this paper is on sensing and communication constraints, the basic approach is applicable to other kinds of formation constraints, including those that arise in cooperative manipulation and mapping tasks. We will show results with our team of car-like robots that show the robots moving towards a goal under such constraints.

2. Problem Definition

Several researchers have considered the multi-robot motion planning problem (see Section 3 for a review). The problem is to find a motion plan for all the robots in a group such that each robot reaches its goal while avoiding collisions with other robots and with the obstacles in the environment. We will extend this problem by defining the *coordinated motion planning problem*, where besides avoiding collisions the robots need to cooperate and maintain formation constraints in order to reach their goals.

Coordinated motion planning problem: Consider a world, $\mathcal{W}$, occupied by a set, $\mathcal{R} = \{R_1, \ldots, R_n\}$, of n robots. The i^{th} robot R_i can be represented by a configuration q_i in the configuration space $\mathcal{C}$. Let $\mathcal{C}^i_{free} \subseteq \mathcal{C}$ denote the free configuration space for R_i. Additionally, let

$\mathcal{C}_{R_i}(\mathcal{R}\backslash R_i, t) \subseteq \mathcal{C}^i_{free}$ denote R_i's valid configuration space imposed by its *formation constraints*. The goal is to steer each robot, R_i, $1 \leq i \leq n$, from a initial configuration q_i^{init} at time $t = 0$ to the goal configuration $q_i^{goal} \in \mathcal{C}^i_{free}$ at some time $t = T > 0$ such that $q_i \in \mathcal{C}_{R_i}(\mathcal{R}\backslash R_i, t)$ $\forall t \in (0, T]$.

Formation constraints are constraints on individual robots induced by the other robots in the team. Thus, $\mathcal{C}_{R_i}(\mathcal{R}\backslash R_i, t)$ depends on the robots' characteristics, their configurations, and also on the nature of the task. Notice that our problem statement differs from the previous definition of multi-robot motion planing problem in the sense that, besides inter-robot collisions, we are adding other kinds of constraints that include, for example, sensor field-of-view constraints and communication range constraints. On the other hand, our definition of multi-robot motion planning is not too different from the definition of robot motion planning for single robots (Latombe, 1991). While the traditional definition considers the problem of moving the robot in a limited free space, where the constraints are induced by the (non-controlled) obstacles in the environment, here part of the valid (free) configuration space for the robots is controlled by the position of the other robots. However, instead of developing a single algorithm for coordinating the motions of all the robots, we develop a decentralized algorithm which allows each robot to choose its motion based on the available free space and the formation constraints.

3. Previous Work

The multi-robot motion planning problem has been addressed with centralized motion planners by a number of groups. The paths are constructed in the composite configuration space $\mathcal{C}_{free} = \mathcal{C}^1_{free} \times \mathcal{C}^2_{free} \times \cdots \times \mathcal{C}^n_{free}$ (Aronov et al., 1998). This approach in general guarantees completeness but its complexity is exponential in the dimension of the composite configuration space (Hopcroft et al., 1984). Other groups have pursued decentralized approaches to motion planning. This generally involves two steps: (1) individual paths are planned independently for each robot; and (2) the paths are merged or combined in a way collisions are avoided. Some authors call these approaches *coordinated path planning* (LaValle and Hutchinson, 1998; Simeon et al., 2002; Guo and Parker, 2002).

Our approach uses potential field controllers based on navigation functions (Rimon and Koditschek, 1992). In a single robot navigating an obstacle field, the navigation function provides a Lyapunov function that guarantees the robot's convergence to the goal. The navigation function can be modified to accommodate unmodeled obstacles or dynamic constraints (Esposito and Kumar, 2002). We will use a similar approach, but in a multi-robot setting, to solve the motion planning problem with formation constraints.

For multi-robot systems, potential fields were used to deploy robots in known environments (Howard et al., 2002). The idea of having artificial potential fields in order to have each robot repelling the others was also used in (Reif and Wang, 1995). These approaches are not directly applicable to our problem because they cannot be easily applied to maintain formation constraints. Further, they do not use the Lyapunov function properties of the potential functions in any meaningful way. In this paper, we present proofs for our methodology which show that besides going to their independent goals the robots also satisfy the formation constraints in the problem.

4. Approach

In this paper we will solve the cooperative motion planning problem defined in Section 2 for a planar world $\mathcal{W} = \mathbb{R}^2$ with sensing and/or communication constraints. We consider a two-level motion planner where the superior level is able to specify a deliberative plan (Arkin, 1998) in terms of previously computed navigation functions for each robot and desired *neighborhood relationships*. The navigation functions are discussed in Section 4.3. The neighborhood relationships are pairwise formation constraints that are formalized in Section 4.2. The second level of the planner is the level we are concerned with. We address the real time modification of the pre-planned functions and the deliberative controller to accommodate the formation constraints. Before we proceed further, we will make some assumptions.

Assumption 1. All robots are identical in terms of geometry, and in terms of capabilities and constraints related to sensing, communication, control, and mobility.

Assumption 2. The sensing and communication devices have a 360^o field of view that can be represented by circles centered at q_i and radius r_i.

Assumption 3. The robots are point robots: $q_i = (x_i, y_i)$.

Assumption 4. The robots are holonomic. For the i^{th} robot, the dynamical model is then given by: $\dot{q}_i = u_i$.

A fifth assumption that we will need for an important proof is an assumption on the individual robot plans. This is introduced in Section 4.3. We will also discuss relaxing the other four assumptions in Sections 4.5.

4.1 Sensing and Communication Networks for Robots

The physical locations of the robots coupled with the characteristics of the hardware and the requirements of the sensing and control algorithms dictate the sensing and communication networks for the group of robots. These networks can be represented by two graphs, G_s and G_c, where the robots themselves are the vertices of both graphs and the flow of information between the vertices are represented by directed edges or arcs. Notice that since sensors and communication devices have different characteristics, the graphs G_s and G_c will have different edge sets. Each graph is represented by a triple $(\mathcal{R}, \mathcal{E}, \mathcal{G})$, where $\mathcal{R}$ is the set of robots, $\mathcal{E} \subseteq \mathcal{R} \times \mathcal{R}$ is the edge set representing communication (or sensing) links among the robots, and $\mathcal{G}$ is the set of constraint functions that describe the conditions under which each link can be maintained. For each element of $\mathcal{E}$ there is a corresponding element in $\mathcal{G}$. Thus, the existence of a communication link between two *neighbor* robots R_i and R_j is represented by the edge $(R_i, R_j) \in \mathcal{E}$ and the function $g(q_i, q_j) \in \mathcal{G}$. Because we consider identical robots and omnidirectional devices, we can restrict our attention to bidirectional graphs. Then, $(R_i, R_j) \in \mathcal{E}$ is equivalent to $(R_j, R_i) \in \mathcal{E}$ and $g(q_i, q_j) \in \mathcal{G}$ is identical to $g(q_j, q_i) \in \mathcal{G}$.

4.2 Formation Constraints

As discussed earlier in the beginning of this section, we assume that the graphs G_c and G_s, and therefore, $\mathcal{E}$, are specified by a higher-level planner. With each edge $(R_i, R_k) \in \mathcal{E}$, we associate a formation constraint for R_i induced by R_k as a inequality of the form $g(q_i, q_k) \leq 0$, where $g(q_i, q_k) \in \mathcal{G}$. In general $g(q_i, q_k)$ could be any convex, differentiable function but given the assumption of omnidirectional devices, and circular robots, all elements of $\mathcal{G}$ will be represented by circles. Then, $g(q_i, q_k) = (x_i - x_k)^2 + (y_i - y_k)^2 - r_k^2$. A constraint is active when $g(q_i, q_k) = \delta_x$, where δ_x is a small negative number that can be thought of as a threshold. The constant δ_x defines then the radius of the circular constraint.

We will consider that a generic robot R_k induces three constraints in R_i. First, we have a hard sensing or communication constraint given by $g(q_i, q_k) \leq \delta_1$, beyond which the connectivity between R_k and R_i is broken. While δ_1 can be taken to be zero, it is best, in order to be robust to sensing errors, to keep it at a small, negative value. Second, we have a soft sensing or communication constraint given by $g(q_i, q_k) \leq \delta_2$. This is assumed to delineate a range within which the performance of the communication or sensing link is optimal. Finally, we have the collision

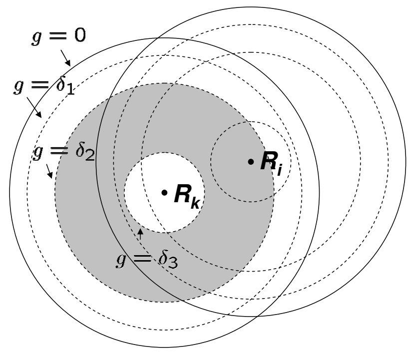

Figure 1. Formation constraints: R_k induces constraints on the position of R_i. If R_1 is inside the circumference defined by $g \leq \delta_1$ (outer dashed circle), connectivity with R_k is guaranteed. The shadowed area defined by $g > \delta_3$ and $g < \delta_2$ is a "safe" configuration space for R_i where collisions are avoided and connectivity is maintained.

constraint $g(q_i, q_k) \geq \delta_3$. Observe that $\delta_3 < \delta_2 < \delta_1 < 0$. Figure 1 shows a picture of R_i constraints induced by R_k. Each neighbor of R_i induces a similar set of constraints. In Sections 4.4, we will define three control modes, one for each region shown in Figure 1.

4.3 Navigation Functions

As discussed in the beginning of this section, a navigation function for solving the non-cooperative problem of steering each individual robot towards the goal while avoiding the static obstacles in the environment is assumed to be available from a higher level planner. Navigation functions are artificial potential fields without local minima (Rimon and Koditschek, 1992). Thus, for a navigation function, V_i, robot R_i input is given by $u_i = -k\nabla V_i$ where ∇V_i is the gradient of V_i. As pointed out in (Esposito and Kumar, 2002), this kind of navigation function can be thought of as a Lyapunov function for the system $\dot{q} = u(q)$, $u(q) = -\nabla V(q)$, because $V(q)$ is positive definite and the value of V is, by definition, always decreasing along system trajectories.

When robots are cooperating, there are formation constraints that force them to navigate near each other and their final goals are reasonably close to each other. In this situation, the gradients of navigation functions for neighboring robots are close to each other. Our fifth assumption is for any pair of cooperating robots, R_i and R_k, $\nabla V_i(q_i) \cong \nabla V_k(q_k)$. Obviously if the robots have goals that are not close to one another and the robots are close to their destinations, this assumption is not valid since their navigation functions will be very different. This fifth assumption is required for the proof of Proposition 2 in Section 4.4.

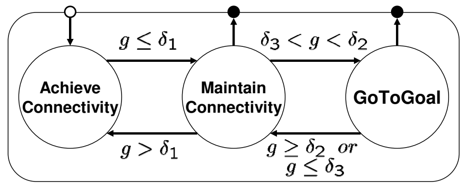

Figure 2. Switched control system with three modes.

4.4 Decentralized Controllers

Our control system is decentralized and implemented using a set of reactive controllers. Each robot switches between these controllers as shown in Figure 2. The switching is governed by activation of constraints that depend on the relative positioning of a robot with respect to its neighbors.

We will develop a controller that allows each robot has up to two assigned neighbors. In other words it assumes the valency of each vertex of G_c and G_s is either one or two. This kind of controller can be very useful because in most of communication and sensing algorithms robots can be used as "routers" between other robots. In this way if all robots have direct communication with two other robots and these robots have connection with at least one different robot then every robot can "communicate" with all other robots in the group.

We will denote the constraints due to R_a, which have the form $g(q_a, q_i) \leq 0$, by g^a and those due to R_b, which have the form $g(q_i, q_b) \leq 0$ by g^b. In the AchieveConnectivity mode each robot tries to move in order to satisfy these constraints without using the navigation function. In other words, the constraints themselves act as potential fields attracting the robots to each other and forcing them into a feasible configuration that satisfies all the constraints. The control input in this mode is:

$$u_i = -k_1 \left(a\nabla g^a + b\nabla g^b \right) \ , \tag{1}$$

where ∇g^x is the gradient of the constraint defined as $\partial g^x / \partial q_i$. ∇g^a is due to R_a and ∇g^b is due to robot R_b. We require these vectors to be normalized and of unit length. The variables a and b assume value 1 or 0 depending whether the constraints are active or not, respectively.

The GoToGoal mode has the following input:

$$u_i = -k_2 v_i \ , \tag{2}$$

where $v_i = \nabla V_i(q_i) / \|\nabla V_i(q_i)\|$ is the normalized gradient vector of the navigation function $V_i(q_i)$. It is a deliberative controller with a pre-planned navigation function that guides the robots toward the goal.

In the MaintainConnectivity mode a robot tries to navigate toward the goal while maintaining the formation constraints. The control input for this state is:

$$u_i = -k_1 \left(a\nabla g^a + b\nabla g^b \right) - k_2 v_i \;, \tag{3}$$

where $k_2 > 2k_1$. In this equation a and b can each be -1, 0, or 1. When $g \leq \delta_3$ the value -1 is assigned. When $\delta_3 > g > \delta_2$, the value 0 is assigned. And when $g \geq \delta_1$, the value 1 is assigned.

These three control laws solve the n problems of individually leading the robots to their goals while guaranteeing the formation constraints are satisfied.

Proposition 1 *If the robots start in a feasible configuration, i.e. a configuration which satisfies all formation constraints, the switched control law represented by (1), (2) and (3) guarantees that those constraints are satisfied during the robots motion.*

The proof for this proposition is similar to the one presented in (Pereira et al., 2002). Space restrictions prevent us from including it in this paper.

Proposition 2 *If the robots start and goal positions are valid configurations and during the motion the gradient of the navigation function of two neighbor robots can be considered the same, the switched control law represented by (1), (2) and (3) leads the robots to their goals.*

The proof of this proposition is straight forward and is based on the fact that $V_i(q_i)$ is a common Lyapunov function for the two modes represented by equations (2) and (3). If the assumption $V_i(q_i) = V_k(q_k)$ is valid, the system will never exit from the MaintainConnectivity mode to the AchieveConnectivity mode. Although the validity of this assumption may be questioned, it is a good assumption when the robots are far away from their respective destinations, as discussed in Section 4.3.

4.5 Extension to Real Robots

Our control laws where derived under Assumptions 1–4. However, most of real robots cannot be represented by at least one of these assumptions and therefore it is natural to ask whether our methodology can be used or not for other kinds of robots.

Assumption 1 cannot be entirely relaxed because our proofs need the constraints for both neighbors robots be active at the same time and

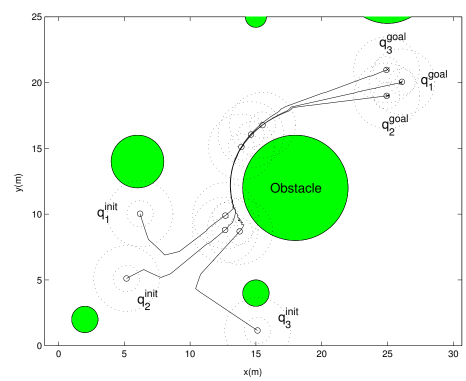

Figure 3. Three holonomic robots navigating to their independent goals while maintain communication constraints.

then, they need to be identical. In this way it is important that the robots have the same sensor and communication characteristics. If they don't, a very good approximation is using the constraints related to the poorest sensor/communication device in the pair of neighbor robots.

In the definition of the control laws, we never explicitly use the fact that sensor constraints are circular. Assumption 2 is only used in order to facilitate the problem understanding. Instead, we only require that each $g(q_i, q_k)$ is convex and differentiable. Actually, the requirement of differentiability can be relaxed if we use generalized gradients in (1) and (3). The controllers and the proof will remain the same except for the fact that each ∇g would be the gradient of one of the functions that represents the active constraint.

Assumption 3 is easily relaxed if the obstacles are grown of the size of the robots during the navigation function construction. For details see (Latombe, 1991).

Assumption 4 is more difficult to deal with. For non-holonomic robots, u_i, which is a two dimensional vector, can be used as a set-point for controllers that take in account the non-holonomic constraints. Experimental results will show that the methodology works in several cases. A different approach is shown in (Esposito and Kumar, 2002) where dipolar potential fields are used to generate potential fields for nonholonomic systems. This is a promising direction for future research.

5. Simulations

Figure 3 shows three simulated robots navigating to their independent goals while maintaining communication constraints and avoiding each other. Observe that in the initial configuration the constraints are not satisfied and the robots need to move closer to each other. Notice also

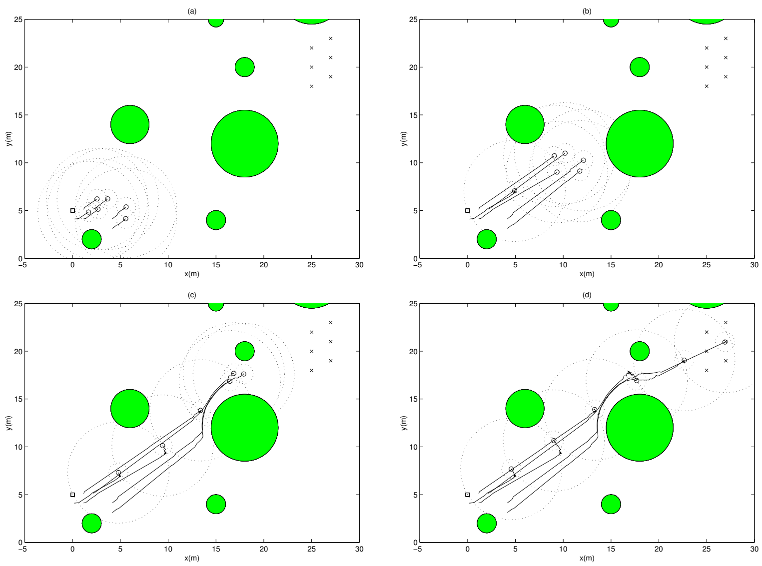

Figure 4. Deploying a mobile sensor network with 6 nodes. Figures (a)–(d) show four snapshots of the same simulation. The first robot maintains communication constraints with the base (square). Roughly speaking only one robot succeeds in its mission of reaching the goal. However, the mission of building a communication link between the goal area and the base is successfully completed.

that the robots do not keep a specific formation during their motion since the constraints determine regions in the robots'configuration space that allow infinite different formations.

Figure 4 shows a simulation that illustrates how local minima introduced by the ACHIEVECONNECTIVITY mode of the controller can be used to deploy a robot network. In this figure the robots have to maintain communication constraints with two other neighbors forming a chain. The first robot, however, has to maintain connectivity with a static base. Consequently, only one of the robots reach the goal but as a side effect a communication link is built between the base and the goal.

6. Experiments

Our platforms are car-like robots equipped with omnidirectional cameras as their primary sensors. The communication among the robots relies on IEEE 802.11 networking. A calibrated overhead camera is used to localize the robots in the environment. Because with this camera we are not able to estimate the robots' orientation, we use communication among the robots in order to construct a complete knowledge of the robots configuration. The communication is essentially used for sensing algorithms but is not used for control or decision making.

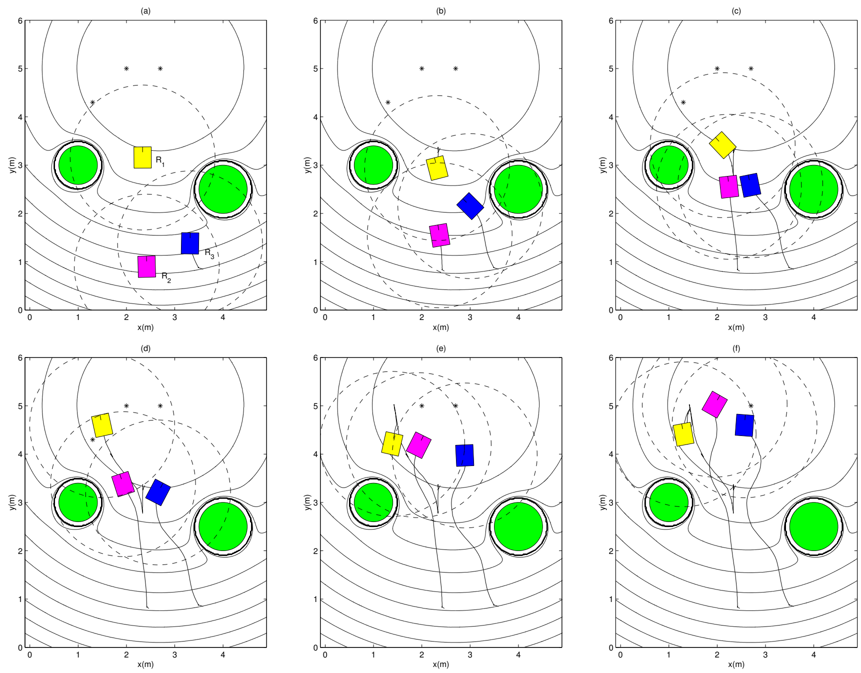

Figure 5. Three robots following their navigation functions while maintain sensing constraints with at least one other robot. Ground truth data is overlaid on the equipotential contours of the navigation function for R_3.

A limitation of the omnidirectional cameras used by the robots is that their resolution decrease with the distance of the objects. At $2\,m$, for instance, the projection of an observed robot in the image plane is only one pixel in size. For this reason three robots were programmed to keep sensing constraints with their neighbors and therefore localize themselves with respect to each other.

Figure 5 shows six snapshots of our experiment. The equipotential contours of the navigation function for R_3 is shown in all snapshots. In this experiment G_s was specified such that R_1 and R_3 are neighbors of R_2 but they are not neighbors of each other. Figure 5(a) shows that R_1 was initialized outside the sensing region of R_2, which was set to be $1.5\,m$. The next snapshot shows that the robots move to satisfy this constraint. Figure 5(c) shows R_2 and R_3 very close two each other. The activation of the avoidance constraints is then followed by a repulsion (Figure 5(d)).

7. Conclusions and Future Work

We developed a suite of decentralized reactive controllers for the cooperative motion control of a group of mobile robots. Our approach is based on the online modification of pre-computed navigation functions in order to satisfy formation constraints. Some proofs of convergence are presented in the case of holonomic robots. Although we have presented results with non-holonomic robots, our proofs are not applicable to these systems.

One of the assumptions in this work has to do with the specification of the communication and sensing network for the reactive controllers. In a current work we are addressing planning methods that will specify the desired edge set for the two graphs, G_c and G_s, which in turn will also precisely define the neighborhood relationships for the team of robots. An important question in this regard has to do with the scalability of such algorithms as the number of robots in the team becomes large.

References

Arkin, R. C. (1998). *Behavior-Based Robotics*. MIT Press, Cambridge, MA.

Aronov, B., de Berg, M., van der Stappen, A. F., Svestka, P., and Vleugels, J. (1998). Motion planning for multiple robots. In *Int'l. Symposium on Computational Geometry*, pages 374–382.

Esposito, J. M. and Kumar, V. (2002). A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 1691–1696.

Guo, Y. and Parker, L. E. (2002). A distributed and optimal motion planning approach for multiple mobile robots. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 2612–2619.

Hopcroft, J. E., Schwartz, J. T., and Sharir, M. (1984). On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "Warehouseman's Problem. *The Int'l Journal of Robotics Research*, 3(4):76–88.

Howard, A., Mataric, M. J., and Sukhame, G. S. (2002). Mobile sensor network deployment using potencial fields: A distributed, scalable solution to the area coverage problem. In *Int'l. Conf. on Distributed Autonomous Robotic Systems*.

Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic, Boston, MA.

LaValle, S. M. and Hutchinson, S. A. (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Trans. on Robotics and Automation*, 14(6):912–925.

Pereira, G. A. S., Kumar, V., and Campos, M. F. M. (2002). Decentralized algorithms for multirobot manipulation via caging. In *Int'l Workshop on Algorithmic Foundations of Robotics*, Nice, France (available online at www.cis.upenn.edu/~guilherm).

Reif, J. and Wang, H. (1995). Social potential fields: A distributed behavioral control for autonoomous robots. In *Int'l Workshop on Algorithmic Foundations of Robotics*, pages 431–459. A. K. Peters, Wellesley, MA.

Rimon, E. and Koditschek, D. E. (1992). Exact robot navigation using artificial potencial functions. *IEEE Trans. on Robotics and Automation*, 8(5):501–517.

Simeon, T., Leroy, S., and Laumond, J.-P. (2002). Path coordination for multiple mobile robots: a resolution complete algorithm. *IEEE Trans. on Robotics and Automation*, 18(1).

Spletzer, J. and Taylor, C. J. (2002). Sensor planning and control in a dynamic environment. In *IEEE Int'l. Conf. on Robotics and Automation*, pages 676–681.